9. Me

Program Name: Me.java **Input File: me.dat**

Hey, it's me! I'm a UIL Computer Science judge, which means I help create programming contest problems. Currently, the problems are sorted in alphabetical order. I had a great idea for our next contest: our problems should be sorted in both alphabetical and difficulty order!

I wrote several problems, and each of the five judges gave a difficulty estimate on each problem. The difficulty of a problem is between 1 and 10 (inclusive), with 1 denoting the easiest problems and 10 denoting the hardest. The difficulty of a problem is the average of the estimates given by the judges.

In some cases, I might have… slightly misjudged problem difficulties, and the list of problems is different when sorted by name and sorted by difficulty. To fix this situation, I can change the first letter (and only the first letter) of the name of each problem. The *weirdness* of changing one letter to another is equal to the square of the distance between the letters in the alphabet. For example, changing a B to an E has weirdness 9, because the letters are 3 apart and the square of 3 is 9. The overall weirdness of a problem set is the sum of the weirdness of changing each name.

I'm only willing to change the first letter of each problem. Can you tell me the minimum weirdness needed to ensure the problem set is sorted in both alphabetical and difficulty order?

For example, say we have 3 problems: Bipul with difficulty 1, Botan with difficulty 2.4, and Tanu with difficulty 1.6. To make these sorted, change Botan to Kotan and Tanu to Kanu. This has overall weirdness 81 + 81 = 162.

Input: The first line of input is T (T <= 10), the number of test cases. Each test case starts with a single integer N (1 <= N <= 50), the number of contest problems written. N lines follow, each containing 6 tokens: the name of the problem followed by the 5 judge scores. Each problem name begins with an uppercase English letter followed by between 1 and 10 (inclusive) lowercase letters. Each difficulty score is between 1 and 10 inclusive.

Output: For each test case, output -1 if there's no way to change the first letters of each problem name so they're sorted in both alphabetical and difficulty order. Otherwise, output the minimum score needed to make the contest sorted in both alphabetical and difficulty order.

Sample input:

```
2
3
Bipul 1 1 1 1 1
Botan 2 3 2 2 3
Tanu 1 1 3 2 1
2
Frad 5 3 4 6 5
Brad 10 8 7 9 9
```

Sample output:

```
Case #1: 162
Case #2: 13
```